



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/779,453	02/09/2001	Christopher C. Tanner	1930.0070002	6696

26111 7590 03/11/2004

STERNE, KESSLER, GOLDSTEIN & FOX PLLC  
1100 NEW YORK AVENUE, N.W.  
WASHINGTON, DC 20005

EXAMINER

YIGDALL, MICHAEL J

ART UNIT	PAPER NUMBER
----------	--------------

2122

DATE MAILED: 03/11/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/779,453

Applicant(s)

TANNER ET AL.

Examiner

Michael J. Yigdal

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 09 February 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-15 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-15 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 09 February 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |  |   |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)  | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)   | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152)             |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)<br>Paper No(s)/Mail Date <u>4</u> . | 6) <input type="checkbox"/> Other: _____  |

### **DETAILED ACTION**

1. Claims 1-15 are pending and have been examined. The priority date considered for the application is 4 August 2000.

#### ***Specification***

2. The abstract of the disclosure is objected to because the abstract must not exceed 150 words. Correction is required. See MPEP § 608.01(b).

#### ***Drawings***

3. Figures 1-4 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g). A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Figures 1-4 are described in the "Background of the Invention" section of the disclosure (see pages 1-7). Figures 1, 3A and 3B, for example, are further shown to illustrate "conventional" procedures (see page 8).

#### ***Claim Rejections - 35 USC § 103***

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-8, 10 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 6,578,197 to Peercy et al. (hereinafter Peercy) in view of U.S. Pat. No. 6,502,238 to Pavan et al. (hereinafter Pavan).

With respect to claim 1, Peercy discloses a method for supporting development of content independent of a run-time platform (see the title, abstract and FIG. 1).

Although Peercy discloses using procedures and functions to define content (see column 6, lines 10-20), Peercy does not expressly disclose the step of:

(a) storing processing blocks that define content.

However, Pavan discloses storing processing blocks that define the content of a program (see column 4, lines 28-37), in order to encapsulate physical device operations or system-level functions that are hidden from the user (see column 4, lines 9-18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the block features taught by Pavan in the system of Peercy, for the purpose of encapsulating low-level device operations.

Although Peercy discloses storing a tree or graph that represents the program (see step 208 in FIG. 2, and FIG. 3), Peercy does not expressly disclose the step of:

(b) storing an application graph that expresses the identity of the stored processing blocks and data connectivity between the stored processing blocks.

However, Pavan further discloses storing an application graph (see FIG. 4) that represents the identity of the blocks and the data connectivity between the blocks (see column 5, lines 8-14, and column 6, lines 18-26), in order to automatically create a distributed application (see column 2, lines 47-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the graph features taught by Pavan in the system of Peercy, for the purpose of automatically creating a distributed application.

Peercy further discloses the limitation of step (b) above whereby the application graph can be traversed by a graphical application platform at run-time to execute appropriate processing blocks on a run-time platform (see steps 210, 212 and 214 in FIG. 2, which show traversing the tree or graph in a graphics platform and then executing the sequence).

With respect to claim 2, Peercy further discloses the limitation wherein the content comprises game content (see column 1, lines 16-23, which shows developing graphics applications for the entertainment industry, i.e. game content; see also column 11, lines 1-2, which further shows using video game cartridges).

With respect to claim 3, Peercy discloses a method for supporting development of content independent of multiple hardware platforms (see the title, abstract and FIG. 1).

Although Peercy discloses using procedures and functions to define content independent of multiple hardware platforms (see column 6, lines 10-20), Peercy does not expressly disclose the step of:

(a) storing processing blocks that define content independent of multiple hardware platforms.

However, Pavan discloses storing processing blocks that define the content of a program (see column 4, lines 28-37), in order to encapsulate physical device operations or system-level functions that are hidden from the user (see column 4, lines 9-18). Note that the user program is

distinct from the system-level program and is thus independent of multiple hardware platforms (see column 7, lines 45-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the block features taught by Pavan in the system of Peercy, for the purpose of encapsulating low-level device operations.

Peercy further discloses the step of:

(b) selecting a target hardware platform from multiple hardware platforms (see step 140 in FIG. 1 and column 6, lines 37-39, which shows targeting a specific hardware platform).

Although Peercy discloses storing a tree or graph that represents the program (see step 208 in FIG. 2, and FIG. 3), Peercy does not expressly disclose the step of:

(c) storing an application graph that expresses the identity of the stored processing blocks and data connectivity between the stored processing blocks based on the selected target hardware platform.

However, Pavan further discloses storing an application graph (see FIG. 4) that represents the identity of the blocks and the data connectivity between the blocks (see column 5, lines 8-14, and column 6, lines 18-26), in order to automatically create a distributed application (see column 2, lines 47-49). Note that the system-level program is based on the selected target hardware platform (see column 6, lines 45-63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the graph features taught by Pavan in the system of Peercy, for the purpose of automatically creating a distributed application.

Peercy further discloses the step of:

(d) traversing the application graph at run-time, including executing appropriate processing blocks on the selected target hardware platform (see steps 210, 212 and 214 in FIG. 2, which show traversing the tree or graph and then executing the sequence).

With respect to claim 4, Peercy further discloses the limitation wherein the content comprises game content (see column 1, lines 16-23, which shows developing graphics applications for the entertainment industry, i.e. game content; see also column 11, lines 1-2, which further shows using video game cartridges), and the multiple hardware platforms include at least one of a game console platform and a personal computer platform (see FIG. 5, which shows a computer platform).

With respect to claim 5, Peercy discloses a game development and run-time system (see the title, abstract and FIG. 1), comprising a graphical application platform that enables a game application to run on any of multiple hardware platforms (see column 6, lines 2-20, which shows a graphics platform that enables an application to run on any of multiple hardware platforms; see also column 1, lines 16-23, which shows developing graphics applications for the entertainment industry, i.e. game applications).

With respect to claim 6, although Peercy discloses that said game application can run on a target hardware platform (see column 6, lines 2-9), as well as a tree or graph that represents the program (see step 208 in FIG. 2, and FIG. 3), Peercy does not expressly disclose an object definition tool that enables a developer to define an application graph.

However, Pavan discloses an object definition program or tool for defining an application graph (see column 6, lines 18-26), in order to simplify programming by making interactions in a distributed application transparent to the developer (see column 7, lines 37-49).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the object definition tool features taught by Pavan in the system of Peercy, for the purpose of simplifying distributed programming.

With respect to claim 7, Peercy/Pavan further discloses the limitation wherein said object definition tool further enables a developer to define objects, object elements, and connections (see Pavan, column 5, line 25 to column 6, lines 8, which shows the developer defining objects along with elements of the objects and the connections between the objects).

With respect to claim 8, Peercy discloses a graphical application platform for leveraging capabilities provided independently in at least one of an application software and a hardware platform (see the title, abstract, and FIG. 1).

Although Peercy discloses using an abstract representation to implement standard features of a graphics API and the capabilities of a hardware platform (see column 5, lines 35-45), Peercy does not expressly disclose:

- (a) an application real-time kernel (ARK).
  - (b) a plurality of standard features implemented as executable blocks of logic; and
  - (c) connections between said blocks that implement data flow between said blocks,
- whereby capabilities of at least one of the application software and the hardware platform can be implemented modularly by adding additional corresponding blocks and connections.



However, Pavan discloses a kernel for the simultaneous, real-time control of applications having multiple inputs and outputs (see column 2, lines 3-14). Pavan further discloses a plurality of standard features implemented as blocks (see column 4, lines 9-18) having interconnections to enable data flow between them (see column 5, lines 8-14). Pavan further discloses that capabilities of the software or hardware platform are implemented modularly by adding blocks and connections (see column 6, lines 45-63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the block features taught by Pavan in the system of Peercy, for the purpose of encapsulating low-level device operations (see Pavan, column 4, lines 9-18) and automatically creating a distributed application (see Pavan, column 2, lines 47-49).

With respect to claim 10, Peercy/Pavan further discloses the limitation wherein said additional blocks implement additional features, said additional features comprising market oriented features (see Pavan, column 4, lines 9-18, which show blocks for implementing functions or features; see also column 1, line 56 to column 2, line 2, which shows control systems having market-oriented features).

With respect to claim 11, Peercy/Pavan further discloses the limitation wherein said additional blocks implement additional features, said additional features comprising application specific features (see Pavan, column 4, line 48 to column 5, line 7, which shows blocks for implementing application-specific features).

Art Unit: 2122

6. Claim 9 is rejected under 35 U.S.C. 103(a) as being unpatentable over Peercy in view of Pavan as applied to claim 8 above, and further in view of U.S. Pat. No. 6,584,489 to Jones et al. (hereinafter Jones).

With respect to claim 9, although Peercy/Pavan discloses resource scheduling and other kernel services (see Pavan, column 6, lines 45-63), Peercy/Pavan does not expressly disclose the limitation wherein said ARK comprises logic that invokes blocks according to a schedule listing the blocks to be executed in each of at least one ARK thread running on at least one central processing unit, dynamically loads and unloads blocks, monitors block execution, and facilitates thread management, memory sharing, mutual exclusion, and synchronization.

However, Jones discloses invoking blocks according to a schedule for one or more threads on one or more processors (see column 6, lines 9-26). Jones further discloses monitoring execution to dynamically load and unload resources or blocks (see column 14, lines 41-48). Jones further discloses managing threads (see column 18, lines 18-39) and resources such as shared memory (see column 5, lines 16-28), as well as mutual exclusion and synchronization (see column 25, lines 41-47).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the scheduling and management features taught by Jones into the system of Peercy/Pavan, for the purpose of enhancing support for real-time applications in distributed environments (see Jones, column 4, lines 57-57).

7. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over Peercy in view of Pavan as applied to claim 8 above, and further in view of U.S. Pat. No. 5,857,106 to Barbour et al. (hereinafter Barbour).

With respect to claim 12, although Peercy/Pavan further discloses the limitation wherein said standard and additional blocks are organized into components (see Pavan, column 4, lines 19-26, which shows primitive or standard blocks and composite or additional blocks; see also Pavan, column 4, lines 2-5, which shows using the concepts of object-oriented programming to organize blocks into components), Peercy/Pavan does not expressly disclose the limitation wherein each component comprises blocks representing alternative implementations of a feature.

However, Barbour discloses libraries or components comprised of modules or blocks that represent alternative implementations of features for specific architectures, in order to provide optimized performance (see column 1, lines 55-65, and column 2, lines 53-61).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to include alternative implementations of features in the system of Peercy/Pavan, as taught by Barbour, for the purpose of providing optimized performance.

8. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over Peercy in view of Pavan in view of Barbour as applied to claim 12 above, and further in view of Jones.

With respect to claim 13, Peercy/Pavan/Barbour further discloses the limitation wherein each of said alternative implementations comprises:

(a) blocks corresponding to said alternative implementation (see Barbour, column 1, lines 55-65, and column 2, lines 53-61).

Although Peercy/Pavan/Barbour discloses identifying the processor and architecture (see Barbour, column 3, lines 29-39), Peercy/Pavan/Barbour does not expressly disclose:

- (b) identification of resources needed by said alternative implementation; and
- (c) identification of resources provided by said alternative implementation.

However, Jones discloses identifying the resources needed by an activity, i.e. by an implementation (see column 9, lines 38-61) and identifying resources provided by an implementation (see column 9, lines 8-14).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to incorporate the resource identification features taught by Jones into the system of Peercy/Pavan/Barbour, for the purpose of enhancing support for real-time applications in distributed environments (see Jones, column 4, lines 57-57).

9. Claims 14 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Peercy in view of Barbour.

With respect to claim 14, Peercy discloses a method of pre-processing a graphics application with respect to a predefined hardware platform (see the title, abstract and FIG. 1).

Although Peercy discloses targeting a specific hardware platform in order to speed up the application (see column 6, lines 2-9), Peercy does not expressly disclose:

- (a) selecting from among a set of alternative implementations of a feature.

However, Barbour discloses selecting a specific implementation of a feature (see column 3, lines 29-39) from among a set of modules or blocks that correspond to alternative

implementations (see column 2, lines 29-39), in order to provide optimized performance (see column 1, lines 55-65).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to select from alternative implementations of a feature in the system of Peercy, as taught by Barbour, for the purpose of providing optimized performance.

Peercy further discloses:

(b) mapping at least one block, corresponding to the selected implementation, to a phase of execution (see column 5, lines 35-45, which shows mapping an abstract expression or block to an intermediate tree representation showing phases of execution);

(c) mapping a phase of execution to a stage of execution (see column 5, lines 46-50, which shows processing the tree to map a phase of execution to a primitive API command, i.e. to a stage of execution);

(d) creating a block execution order list corresponding to the stage of execution (see column 5, lines 46-61, which shows creating a command sequence, i.e. an execution order list);

(e) submitting the stage of execution to an application real time kernel for management of execution of the stage (see column 5, lines 54-56, which shows submitting the command sequence for execution).

With respect to claim 15, Peercy/Barbour further discloses the limitation wherein said step (a) comprises a negotiation process in which resource requirements of each alternative implementation are considered, along with the costs and benefits of variations in such resource requirements, thereby allowing selection of an implementation (see Peercy, column 5, lines 46-

Art Unit: 2122

61, which shows considering the costs of an implementation in terms of execution time and resource requirements and selecting an efficient sequence).

10. Claims 1-15 are also rejected under 35 U.S.C. 103(a) as being unpatentable over applicant's own admitted prior art (see pages 3-6).

Applicant discloses several prior art systems for graphical application development and graphical run-time environments, including those intended for game content (see pages 3-6). It is noted that applicant describes several deficiencies of the prior art systems, such as weak extension models, limited code reuse, difficulties in integration, and duplication of application state (see pages 4-5). However, any features of the present invention directed to overcoming such deficiencies are not expressly recited in the claims.

### ***Conclusion***

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. U.S. Pat. No. 5,815,415 to Bentley et al. discloses a graphical modeling system comprising an application framework and a kernel for execution on more than one platform.

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Michael J. Yigdall whose telephone number is (703) 305-0352. The examiner can normally be reached on Monday through Friday from 8:00am to 4:30pm.

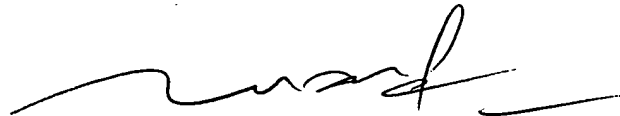
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

MY

Michael J. Yigdall  
Examiner  
Art Unit 2122

mjy  
March 4, 2004



**TUAN DAM**  
**SUPERVISORY PATENT EXAMINER**